

Workshop on Software Engineering for High Performance Computing System (HPCS) Applications

Philip Johnson
Information and Computer Sciences
University of Hawaii
johnson@hawaii.edu

1. Introduction

High performance computing systems are used to develop software for wide variety of domains including nuclear physics, crash simulation, satellite data processing, fluid dynamics, climate modelling, bioinformatics, and financial modelling. The TOP500 website (<http://www.top500.org/>) lists the top 500 high performance computing systems along with their specifications and owners. The diversity of government, scientific, and commercial organizations present on this list illustrates the growing prevalence and impact of HPCS applications on modern society.

Recent initiatives in the HPCS community, such as the DARPA High Productivity Computing Systems program [1] and the Workshop on the Roadmap for the Revitalization of High-End Computing [2], recognize that dramatic increases in low-level HPCS benchmarks of processor speed and memory access times do not necessarily translate into high-level increases in actual development productivity. While the machines are getting faster, the developer effort required to fully exploit these advances can be prohibitive. There is an emerging movement within the HPCS community to define new ways of measuring high performance computing systems, ways which take into account not only the low-level hardware components, but the higher-level productivity costs associated with producing usable HPCS applications. This movement creates an opportunity for the software engineering community to apply our techniques and knowledge to a new and important application domain.

Furthermore, the design, implementation, development, and maintenance of HPCS software systems can differ in significant ways from the systems and development processes more typically studied by the software engineering community.

First, the requirements often include conformance to sophisticated mathematical models. Indeed, requirements may often take the form of an executable model in a system such as Mathematica, and the implementation involves porting to the HPCS.

Second, the software development process, or "workflow" for HPCS application development may differ profoundly from traditional software engineering processes. For example, one scientific computing workflow, dubbed the "lone researcher", involves a single scientist developing a system to test a hypothesis. Once the system runs correctly once and returns its results, the scientist has no further need of the system. This contrasts with standard software engineering lifecycle models, in which the useful life of the software is expected to begin, not end, after the first correct execution.

Finally, "usability" in the context of HPCS application development may revolve around optimization to the machine architecture so that computations complete in a reasonable amount of time. The effort and resources involved in such optimization may exceed initial development of the algorithm.

2. References

- [1] DARPA High Productivity Computing Systems Program, <http://www.highproductivity.org/>.
- [2] The Roadmap for the Revitalization of High-End Computing, Daniel Reed, Editor, Computing Research Association, 2003.